Bankzie Chatbot Requirements Specification

Himasai E

5.1.3.1 Sending a question

5.2 Activity Diagram

1. Introduction

1.1 Purpose of Document

This document will provide all of the requirements for the project Bankzie Chatbot. It will serve as a reference for developers and customers during the development of the final version of the system.

1.1 Project Scope

Bankzie Chatbot is an AI bank chatbot that receives questions from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, when it receives the question what is my balance, it will give a response “The balance is 90000.”

The main objective is creating a Web API, and sample web, mobile, and text messaging interfaces that demonstrate the use of the API.

The goal is to provide Hungiton and faculty a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate Bankzie Chatbot into their projects.

1.1 Overview of Document

1. **Introduction:** Provide an overview of the application, explain the objectives and goal of the project and describe the document structure.

2. **Overall Description:** Provide the specification of the system model, the classes model and the main constraints.

3. **Functional Requirements:** Provide the analysis of the requirements by feature.

4.**Non functional requirements:** Provide some other constraints that affect performance, safety and security.

4. **Use Cases:** Provide possible scenarios where the user interacts with the Web API and sample applications.

3. Description

3.1 Product Perspective

Most of the search engines today, like Google, use a system (The Pagerank Algorithm) to rank different web pages. When a user enters a query, the query is interpreted as keywords and the system returns a list of highest ranked web pages which may have the answer to the query. Then the user must go through the list of webpages to find the answer they are looking for. Chatbot bankzie, however, will try to understand the query and provide a definitive answer.

There will be four main units to the system working together to understand the question and return an appropriate answer:

- Generic question construction - capable of taking a natural language question and making it more generic.
- Generic answer construction - capable of taking a generic question template and providing a generic answer template.
- Generic answer population - capable of taking a generic answer template and populating it with information from the database to form an answer.
- Information extraction - capable of finding information through structured or unstructured websites, and storing that information in a database.

3.2 Product Features

The major features for bankzie Chatbot will be the following:

- **Web API:** An API call will include a question in the form of a query string url parameter and the service will reply in JSON.
- **Natural Language Processing:** The system will take in questions written in standard English.
- **Natural Language Responses:** The answer to the question will be written in standard and understandable English.

Information Extraction: There will be a database containing all the information needed, populated using information extraction techniques

3.3 User Classes and Characteristics

The two classes of users for this system are described below:

- **API users**

  API users consist of application developers who want to incorporate bankzie Chatbot API into bank customer service

- **Mobile app/Web app/SMS users**

  These users consist of non-technical users who want to get answers for their questions. These users ask questions and get answers with mobile, web, or text messaging interfaces for their bank accounts

3.4 Constraints

3.4.1 Limited Question Scope

Creating a chatbot able to answer every single question about bank is not possible to implement with current technology and within the duration of the project, so the system will be able to answer questions about limited topics.

3.4.2 Language

The system will only support questions in standard English.

3.5 Assumptions and Dependencies

Keras is a library for creating and using neural networks. It should provide us with all the functionality we need, however if it is in someway deficient, then it will be replaced with a different library.

BeautifulSoup is a library for parsing HTML documents. It should be all we need to extract text from a webpage, but may be replaced if necessary.

We will develop the project using Python and MySQL database.

3.6 Requirements Apportioning

Each feature will be assigned an important value. The project will be complete if all the features of Priority 1 and at least 50% of features of Priority 2 are implemented. No Priority 3 requirements are necessary.

Priority	Meaning
1	Essential, the project will not work without this feature. This feature will be implemented.
2	Importantly, the scope of the project will be significantly hindered without this feature. This feature will likely be implemented.
3	Desired, this feature complements the core functionality. This feature will be implemented, time allowing.

4. Functional Requirements

4.1 API Calls

4.1.1 Client Responsibilities

R4.1.1.1 The client will send a GET **Priority 1**

R4.1.1.2 The client will specify the header Content-Type: application/json in their requests as convention. **Priority 1**

R4.1.1.3 A valid API query is a single URL parameter containing one sentence that is a question in standard English. **Priority 1**

R4.1.1.4 The server will reply with either data or an error, see R4.1.3.5. The client will be able to parse the JSON and determine if there was an error. **Priority 1**

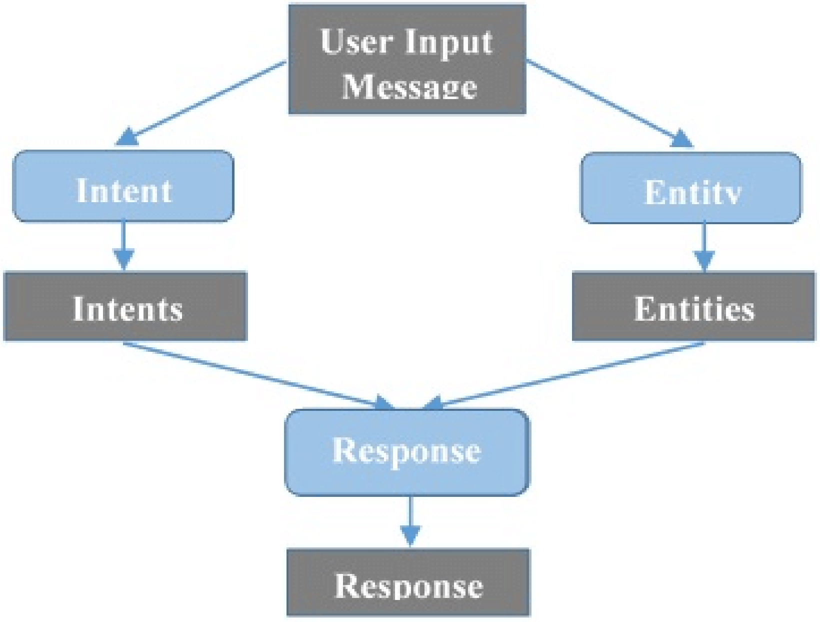
4.1.2 Server Responsibilities

R4.1.2.1 The server will send all API data in JSON response documents with the header Content-Type: application/json. **Priority 1**

R4.1.2.2 The server will respond with a 200 OK status code if a request has the header Content-Type: application/json and is a valid API query. **Priority 1**

R4.1.2.3 The server will respond with a 400 Bad Request status code if a request does not specify the header Content-Type: application/json OR is a malformed API query. **Priority 1**

4.1.3 Response Document Structure



4.2 Generic Question Construction

4.2.1 Input & Output Format

R4.2.1.1 This unit will receive a text string from the URL parameter.**Priority 1**

R4.2.1.2 This unit will identify important words in the sentence and replace them with generic representations preceded by an escape character. (example: “PISB” becomes “$building”) **Priority 1**

R4.2.1.3 This unit will output the sentence as a string, as described in R4.2.1.2. **Priority 1**

R4.2.1.4 This unit will output a map of generic representations to the words they replaced. **Priority 1**

4.2.2 List of Generic Representations

R4.2.2.1 This unit will have a list of generic words related specifically to potential queries (examples: building, professor). **Priority 1**

4.2.3 Error Handling

R4.2.3.1 An error during this process means there was a problem parsing the sentence and creating the generic question. In this case, return a message such as “Sorry, I didn’t understand that.” **Priority 1**

4.5 Information Extraction & Database

4.5.1 Database

R4.5.1.1 A MySQL database will be used to store all information required to answer questions. Priority 1

R4.5.1.2 The database is populated by the information extraction unit before the rest of the system is available, so that all information is readily accessible. Priority 1

R4.5.1.3 The database will use the generic representations from R4.2.2.1 as table names or column headers for easier retrieval of data. Priority **1**

R4.5.1.4 The database will be updated periodically and the API will be unavailable during the update. Priority 1

4.5.2 Structured Input

These are highly organized data sources, such that including the data into our database is simple. Examples: Databases

R4.5.2.1 Structural data sources will have their data stored in our database. Priority 1

4.5.3 Semi-structured & Unstructured Input

Semi-structured data sources are data sources with some organization, but the structure is not rigid enough to assure easy extraction of data. Example: table on a website.

Unstructured data is data with no organization, so extracting information is very hard to do programmatically. Example: A paragraph.

Extraction of information from semi-structured and unstructured data sources can be handled in 3 possible ways:

R4.5.3.1 For data with enough structuring, web scraping will be used to programmatically extract all the data needed and store it in our database. Priority 1

R4.5.3.2 For totally unstructured information, AI libraries will be used to programmatically extract and store any information possible. Priority 2

R4.5.3.3 Data that is especially hard to extract, for whatever reason, will be manually extracted and added to the database by a developer.Priority 3

4.6 Supported Question Topics

The Web API will only handle questions from following topics without unexpected error:

R4.6.1.1 Bank account details **Priority1**

R4.6.1.2 Bank account details **Priority 1**

R4.6.1.3 Bank pin change **Priority 1**

R4.6.1.4 Bank loan emi **Priority1**

R4.6.1.5 Credit report **Priority 2**

R4.6.1.6 credit card details **Priority 2**

R4.6.1.7 Money market values,IRAs **Priority 2**

R4.6.1.8 Saving account openings/Closings **Priority 2**

R4.6.1.9 branch locations **Priority 2**

R4.6.1.10 bank locations directions**Priority 2**

R4.6.1.11 dispute transaction **Priority 2**

R4.6.1.12 raise a complainant **Priority 3**

R4.6.1.13 track complaint status **Priority 3**

4.7 User Interfaces

4.7.1 Website and mobile application GUI

R4.7.1.1 The GUI will have a textbox that will accept inputs from a keyboard. **Priority 1**

R4.7.1.2 Text box will originally contain a suggestive text question, to guide the user to the format of an appropriate question. **Priority 1**

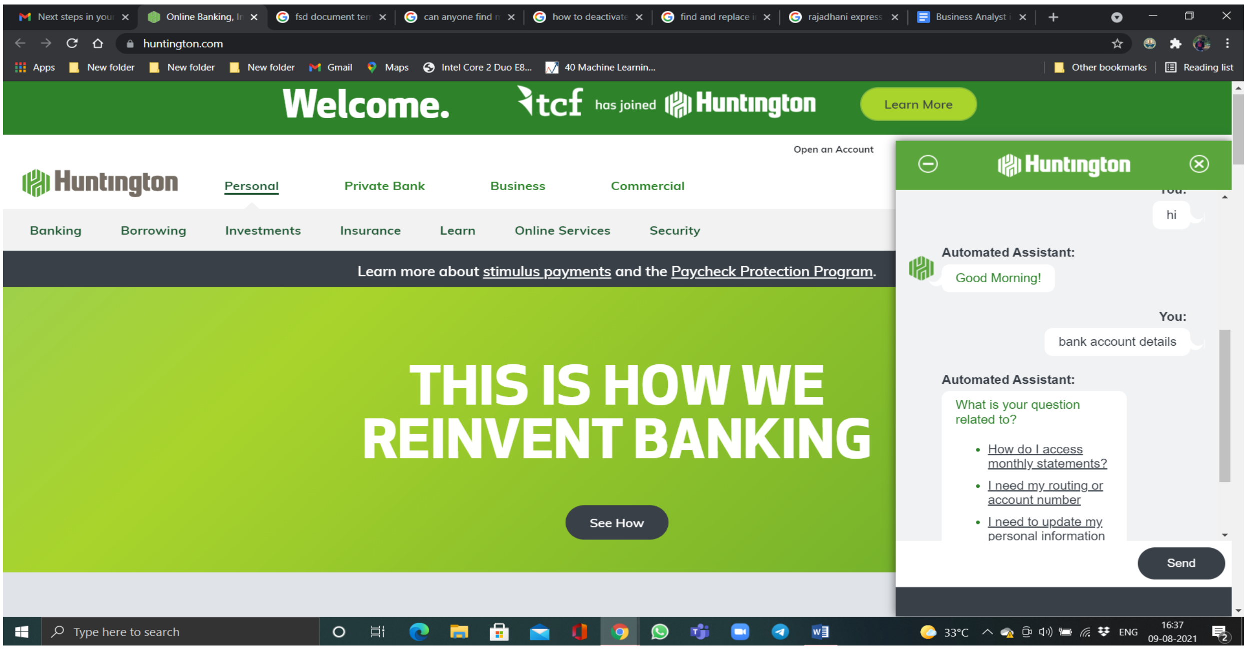
R4.7.1.3 The GUI will have a "Send" button which sends text from the textbox to the API when clicked. **Priority 1**

R4.7.1.4 The GUI will have a chat window displaying questions sent to the system and responses from the API. **Priority 1**

R4.7.1.5 The chat window will contain all questions and answers from the current session, with a scroll bar if all messages can't fit on the screen.**Priority 1**

R4.7.1.6 If there is a network issue, the chat window will display an error message. **Priority 1**

USER INTERFACE:



You:
hi

Automated Assistant:
Good Morning!

You:
bank account details

Automated Assistant:
What is your question related to?

- How do I access monthly statements?
- I need my routing or account number
- I need to update my personal information

3. Non-functional Requirements

3.1 API

5.1.1 Modularity

R5.1.1.1 The system will be designed in such a way that the algorithms for the four main units will be able to be easily swapped out. **Priority 1**

5.1.2 Accuracy

R5.1.2.1 The overall accuracy of the Web API's response will be measured using a developer-made testing set. **Priority 1**

R5.1.2.2 The overall accuracy is calculated by dividing total number of correct answers by the number of questions asked. **Priority 1**

R5.1.2.3 The accuracy of the Generic Question Construction part will be close to 80%. **Priority 2**

R5.1.2.4 The accuracy of the Generic Answer Construction unit will be close to 70%. **Priority 2**

R5.1.2.5 The accuracy of the Generic Answer Population unit will be close to 70%. **Priority 2**

The accuracy for each supported topic will be as follows:

R5.1.2.6 Drexel facilities' locations and schedules will have accuracy greater than 70% **Priority 2**

R5.1.2.7 Drexel staff's office locations, contact information, and positions will have accuracy greater than 70% **Priority 2**

R5.1.2.8 Drexel policies including academics, admissions, information technology etc. will have accuracy greater than 70%. **Priority 2**

R5.1.2.9 On-campus dining locations, hours, food types, etc. will have accuracy greater than 50%. **Priority 2**

R5.1.2.10 Food trucks' general locations, hours, and food types on Drexel Campus will have accuracy greater than 40%. **Priority 3**

R5.1.2.11 Official events listed on Drexel website, location and hours will have accuracy greater than 40%. **Priority 3**

5.1.3 Fast Response

R5.1.3.1 The average time for the server to respond, over the question testing set, will be less than or equal to 2 seconds. **Priority 2**

5.1.4 Security

R5.1.4.1 The connection between the Web API and the programs will use HTTPS, for security. **Priority 3**

5.2 Web interface/Mobile application

5.2.1 Ease of Use

R5.2.1.1 A new user will make less than 3 mistakes in 5 minutes after 5 minutes of use. **Priority 1**

6. Use Cases

6.1 Use Case Flow

6.1.1 API

Precondition: The server that is running the API is online.

Main Flow: The user sends their question as a URL parameter to our API's url.

Postcondition: The user receives the answer to their question in JSON.

6.1.2 Web/mobile application

6.1.2.2 Entering a question

Preconditions: The user starts the mobile application or web application.

Main Flow: The user enters the question in the text box

Postcondition: The text box will show the question entered.

6.1.2.2 Sending a question

Precondition: Some texts exist in the Text Field box.

Main Flow: User clicks the send button

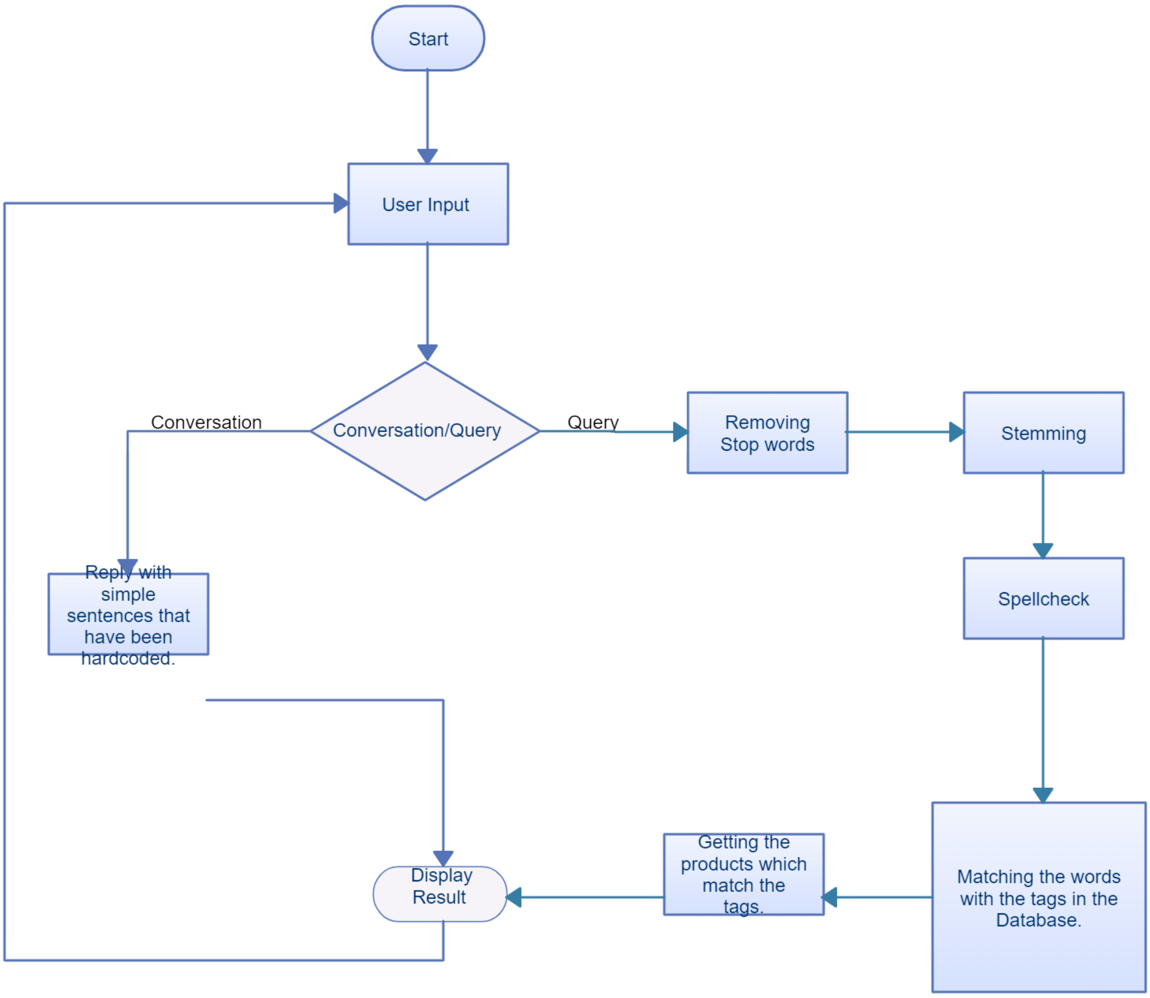
Postcondition: The texts in the Text Field box appear in the chat window, and the box is cleared out. After sometime, some texts generated by the software appear in the chat window.

6.1.3 SMS application

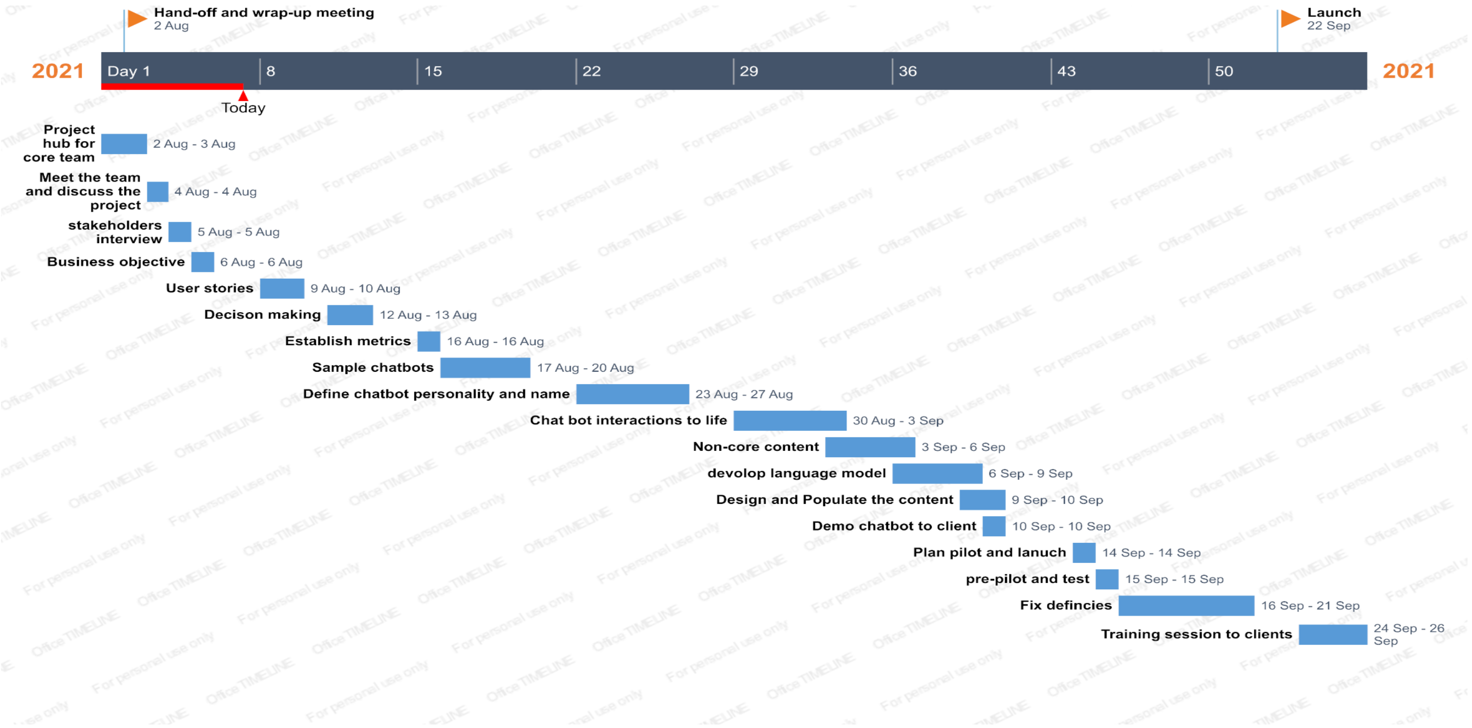
6.1.3.1 Sending a question

Precondition: User can send and receive SMS messages **Main Flow**: Create and send a text message to the designated Drexel Chatbot number

Postcondition: The answer to the user’s question is sent as a SMS to the number that sent the question.


Start
User Input
Conversation
Conversation/Query
Query
Removing Stop words
Stemming
Reply with simple sentences that have been hardcoded.
Spellcheck
Getting the products which match the tags.
Display Result
Matching the words with the tags in the Database.

WSD:


Hand-off and wrap-up meeting
2 Aug
Launch
22 Sep
2021
Day 1
8
15
22
29
36
43
50
2021
Today
Project hub for core team
2 Aug - 3 Aug
Meet the team and discuss the project
4 Aug - 4 Aug
stakeholders interview
5 Aug - 5 Aug
Business objective
6 Aug - 6 Aug
User stories
9 Aug - 10 Aug
Decison making
12 Aug - 13 Aug
Establish metrics
16 Aug - 16 Aug
Sample chatbots
17 Aug - 20 Aug
Define chatbot personality and name
23 Aug - 27 Aug
Chat bot interactions to life
30 Aug - 3 Sep
Non-core content
3 Sep - 6 Sep
devolop language model
6 Sep - 9 Sep
Design and Populate the content
9 Sep - 10 Sep
Demo chatbot to client
10 Sep - 10 Sep
Plan pilot and lanuch
14 Sep - 14 Sep
pre-pilot and test
15 Sep - 15 Sep
Fix defincies
16 Sep - 21 Sep
Training session to clients
24 Sep - 26 Sep

Bank Chatbot Project Plan (12 weeks)

This project plan provides the work breakdown structure that our expert chatbot consultants use to deliver a new chatbot. We add, adjust or delete tasks depending on the needs of each client. Project length typically ranges from 4 to 16 weeks depending on complexity.

Task	Goal	Deliverable	Week	Notes
Project start-up & management			**1+**	
Kick-off meeting	Start the project, meet the team, discuss project plan	Meeting notes	1	MILESTONE
Define team	Gain a basic understanding of who will be doing what so that the project can get started	List of people and roles	1	
Create project hub for core team	Establish the method to be used for project communication and sharing deliverables	Successful access for each member of core team	1	
Hold weekly status meetings & prepare weekly status reports	Keep all stakeholders informed of progress, next steps, critical issues, budget used and remaining	Weekly status reports	2-12	

Task	Goal	Deliverable	Week	Notes
Planning Phase			**2**	
Interview stakeholders to understand needs	Understand needs and priorities of individual stakeholders	Interview notes	2	1-on-1 interviews often yield more insights than group sessions
Determine business objectives through group meeting / workshop	Identify and prioritize business objectives that have approval of stakeholders and budget holder	List of prioritized objectives	2	Techniques used could include a "Hopes & Fears" workshop, co-created process mapping, sticky-note brainstorming, affinity mapping, and dot-voting
Translate objectives into user stories	Capture objectives in a Who/What/Why format that describes how the chatbot will deliver on the objectives so that core team can easily answer, "Have we delivered on this objective?"	List of user stories	2	
Formalize decisionmaking	Determine who is responsible for each aspect of the chatbot and when decisions will be made so that future decisionmaking is efficient	Team definition. RACI. Meeting schedule. Meeting agenda template	2	
Establish meaningful metrics	Identify a few simple metrics that will track how well the chatbot is delivering on business objectives	3 to 5 key metrics that measure the key business objectives	2	

Task	Goal	Deliverable	Week	Notes
Content Phase			**3-6**	
Write sample chatbot interactions	Clearly document the types of interactions the chatbot supports and the way the interactions feel	A document with a series of supported chatbot interactions	3-4	MILESTONE. This document varies widely depending on the chatbot objectives. A document for a FAQ-type chatbot may contain a list of question and answer pairs. A document for a people-finder chatbot may demonstrate different ways a user might ask for a person. A script for a data collection chatbot may contain dialog for an interview
Gather content	Gather all the content that will be delivered via the chatbot	Repository of content (or spreadsheet of content sources) that will be used to populate chatbot	3-4	Depending on chatbot objectives, this step could be very short or very long. Consultants may need to be granted access to appropriate systems to help gather content
Define chatbot personality	Define the voice or tone that the chatbot will take	A few sentences describing the personality of the chatbot. Examples of words, phrases and punctuation characteristic of the personality	3-4	It's easy to go overboard on this step. For an enterprise chatbot, it's more important that the chatbot is *useful* and *usable*. A delightful personality should be viewed as a nice-to-have
Choose the name	A name that can be used in branding and in sample scripts	An agreed-upon name for the chatbot	3-4	Some organizations engage employees in the excitement of the new chatbot by inviting their help in choosing the name
Develop non-core content including pleasantries and failures	Support successful conversation that surrounds core content, including pleasantries, edge-cases, and failures. On failure, redirect user back to supported conversation	Document with responses for pleasantries, edge-cases, failures	5	
Develop language model	For chatbots that support natural language, develop initial language model so content population can commence	Initial language model with intents, entities and sample utterances	5-6	Some chatbots are completely button-based and do not require a language model, greatly reducing deployment time
Edit content for chatbot use	Adjust content so it is appropriate for delivery via chatbot. Shorten it and add personality	Spreadsheet or other repository of edited content	5-6	Depending on chatbot objectives, this step could be very short or very long
Populate content	Get content into the chatbot	Bot responds with content	5-6	Depending on chatbot objectives, this step could be very short or very long
Design icon and logotype	Build a simple visual identity for the chatbot that matches the organization brand	Icon and logotype for chatbot	5-6	

Task	Goal	Deliverable	Week	Notes
Technology Phase			**5–9**	
Identify and describe new functionality necessary to make chatbot interactions come to life	Provide dev team with functional specifications so they can estimate and start on development	Series of tasks for development team to complete	4-5	Many chatbots require custom development to interface with other systems or to perform functions that are not available by default in the chosen chatbot platform
Develop custom integrations and functionality	Get chatbot to work as specified	Custom features or integrations	5-9	This step could be much shorter or longer depending on the integrations or functionality that is to be built
Configure bot on selected channel(s)	Get an official, approved bot on each selected channel	Bot on each selected channel responds to input	6	For some channels, IT assistance or authorization may be required
Apply the look	Implement the design on channels	Bot brand appears in appropriate places on channel	6	

Demo Chatbot Alpha to client	Provide client with a first look	Controlled demo to client	8	MILESTONE
Task	**Goal**	**Deliverable**	**Week**	**Notes**
Pilot & Launch Phase			**8–12**	
Plan pilot and launch	Plan activities to support the pilot, launch and adoption of the new chatbot	Pilot and launch plan	8	
Conduct pre-pilot review	Review the user stories and ensure the chatbot delivers on each one.	Punchlist with outstanding items that must be resolved prior to pilot	8-10	
Test & revise content	Adjust and improve bot interaction based on core team feedback	Mini-pilot to core team. Punchlist with outstanding items that must be resolved prior to pilot.	8-10	Core team available to test and report issues.
Fix deficiencies	Fix issues identified on punchlists	Chatbot ready for pilot	9-10	
Pilot / End user introduction	Gradually introduce the chatbot to end users and familiarize them with how to start using it. Refine content and language model based on transcripts.	Chatbot Beta. Training sessions / bot orientations. Updated chatbot.	10-12	MILESTONE. For a button-based bot, pilot can be brief. For a bot with natural language understanding, pilot should gradually expand over many weeks or months. Transcripts must be carefully reviewed to revise and improve language model
Launch!	Inform the entire organization that the chatbot is ready!	Chatbot Final. Live chatbot, accessible by all users	12+	Senior leader will send an email or other communication announcing launch of chatbot.
Revise content	Maintain a high-performing chatbot	Continuously improving chatbot	Ongoing	
Task	**Goal**	**Deliverable**	**Week**	**Notes**
Project wrap-up			**12**	
Prepare documentation or hold training session(s).	Prepare client to handle ongoing chatbot administration	Client-specific documentation or training session(s)	12	
Hand-off and wrap-up meeting	Pronounce the project complete. Identify what went well, what went poorly. Make recommendations for what should be done next.	Project debrief report	12	MILESTONE